

**Universidad Complutense de Madrid**

**Facultad de Informática**



**Implementación de una aplicación móvil iOS  
para la gestión de pacientes con enfermedades  
cardiovasculares**

Alumno **Juan Carlos Martín Sánchez**

Director de proyecto **Jose Ignacio Hidalgo Pérez**

Trabajo de Fin de Grado

Grado en Ingeniería de Computadores

Septiembre de 2018



# Índice general

<b>Resumen</b>	<b>5</b>
<b>Conclusiones</b>	<b>7</b>
<b>Abstract</b>	<b>9</b>
<b>Conclusions</b>	<b>11</b>
<b>1. Introducción</b>	<b>13</b>
1.1. Motivación . . . . .	13
1.2. Objetivos . . . . .	14
1.3. Estado del arte . . . . .	14
<b>2. Especificación de la aplicación</b>	<b>17</b>
2.1. Pantalla de selección del tipo de registro . . . . .	17
2.2. Pantalla de login convencional . . . . .	19
2.3. Login a través de Facebook . . . . .	21
2.4. Pantalla principal de la aplicación . . . . .	22
2.5. Calculador del índice de masa corporal . . . . .	23
2.6. Calculador de la hipertensión arterial media . . . . .	24
2.7. Cálculo del tabaquismo . . . . .	27
2.8. Cálculo del colesterol . . . . .	29
2.9. Calculador de bolos . . . . .	31
2.9.1. Introducción al calculador de bolos . . . . .	31
2.9.2. Datos que deben ser proporcionados por el usuario . . . . .	31
2.9.3. Cálculo del bolo prandial . . . . .	32
2.9.4. Bolo corrector para después de la comida . . . . .	34
<b>3. Arquitectura de la aplicación</b>	<b>35</b>
3.1. Aplicación iOS . . . . .	36
3.2. Base de Datos . . . . .	36

3.3. Servidor . . . . .	36
<b>4. Tecnologías empleadas</b>	<b>37</b>
4.1. Aplicación iOS . . . . .	37
4.2. Base de datos . . . . .	37
<b>5. Modelo de datos</b>	<b>39</b>
<b>Bibliografía</b>	<b>41</b>
<b>Agradecimientos</b>	<b>43</b>

# Resumen

Las aplicaciones móviles y su uso han ido aumentando con el paso de los años, siendo así una de las principales herramientas de uso en el mundo tal y como los conocemos, estas, se encuentran implicadas en el desarrollo diario la mayoría de profesionales.

Este proyecto fin de Grado presenta una aplicación móvil para dispositivos iOS enfocada en el ámbito de la medicina, y las metas es ofrecer a lo usuarios de la misma, información acerca de los factores relativos al riesgo cardiovascular y a calcular bolos de insulina en pacientes con diabetes. Para poder conseguir este objetivo he diseñado una interfaz moderna, sencilla y fácil de usar, el usuario podrá de forma ágil y sencilla calcular su riesgo vascular y la cantidad de insulina que tiene que consumir tanto antes como después de las comidas que realice.

Esta aplicación está diseñada para que la gestionen los propios usuarios, permitiéndoles almacenar en la base de datos los datos que permite la aplicación concernientes al:

- IMC: Índice de masa corporal
- HTA: Hipertensión arterial media.
- Tabaquismo
- LDL: Lipoproteína de baja densidad
- Bolo prandial
- Bolo corrector



# Conclusiones

Uno de los objetivos más importantes de este trabajo de Fin de Grado era encontrar algunos de los límites entre las tareas que realizan los profesionales en la actividad sanitaria y la evolución de las aplicaciones móviles.

Con respecto a las aplicaciones móviles y a las herramientas informáticas, la medicina ha experimentado una de las evoluciones más lentas. Por tanto, el proyecto que se ha realizado permite ir un paso más allá en la evolución de este campo tan importante para todos.

En este aspecto, la medicina se ha sucedido una evolución lenta en lo que se refiere a esta revolución hacia la automatización de los procesos. Por todo ello, el presente proyecto toma como objetivo la innovación en este campo.

Desarrollando esta aplicación se ha podido observar el gran movimiento y avance que se sufre en las plataformas tecnológicas en cuestión de meses y el esfuerzo del desarrollador que eso supone.

En el transcurso del curso, se han sucedido una serie de actualizaciones en la versión 9 de Xcode, pero estas actualizaciones han sido necesarias para adaptarse a las necesidades que tenían los desarrolladores de iOS con la llegada del iPhoneX y su pantalla. He podido conocer esta herramienta desde cerca y puedo afirmar sin ninguna duda que es el mejor SDK con el que he trabajado ya que es bastante intuitivo y te permite crear ventanas de una manera agradable y todo ello sin tener que escribir ni una sola línea de código.

No obstante, todo no fueron partes positivas para esta herramienta ya que a principio de año sufrió un bug en el que el simulador de iOS se relentizaba al deslizar las ventanas, bug que no pudieron solucionar hasta un mes más tarde con una nueva versión.

Pese a las dificultades que se han sucedido durante el curso con xCode, sobre todo al comienzo del curso, ya que no había mucha información todavía para Swift4, la realización de este trabajo y la experiencia que he obtenido usándolo, me han ofrecido conocimientos de gran utilidad, incluso para abrirme alguna puerta en el mundo laboral.





# Abstract

Mobile applications and their use have been increasing over the years, being one of the main tools of use in the world as we know them, these are involved in the daily development of most professionals.

This end of degree project presents a mobile application for iOS devices focused on the field of medicine, and the scopes is to offer users of it, information about the factors related to cardiovascular risk and calculate insulin boluses in patients with diabetes . In order to achieve this goal I have designed a modern, simple and easy to use interface, the user can easily and easily calculate your vascular risk and the amount of insulin you have to consume both before and after meals.

This application is designed to be managed by the users themselves, allowing them to store in the database the data that the application allows concerning:

- Body mass index
- High blood pressure
- Smoking
- Cholestherol
- Prandial bolus
- Corrective bolus



# Conclusions

One of the most important objectives of this Final Degree project was to find some of the limits between the tasks carried out by professionals in healthcare and the evolution of mobile applications.

With respect to mobile applications and computer tools, medicine has experienced one of the slowest evolutions. Therefore, the project that has been made allows us to go a step further in the evolution of this field that is so important for all.

In this regard, medicine has undergone a slow evolution in what is referred to as this revolution towards the automation of processes. For all this, the present The project aims at innovation in this field.

Developing this application it has been possible to observe the great movement and advance that is suffered in the technological platforms in a matter of months and the effort of the developer that that supposes.

During the course, there have been a series of updates in version 9 of Xcode, but these updates have been necessary to adapt to the needs of iOS developers with the arrival of the iPhoneX and its screen. This tool from near and I can say without any doubt that it is the best SDK with which I have worked since it is quite intuitive and allows you to create windows in a pleasant way and all without having to write a single line of code.

However, everything was not positive for this tool since at the beginning of the year it suffered a bug in which the iOS simulator slowed down when sliding the windows, a bug that could not be solved until a month later with a new version.

Despite the difficulties that have occurred during the course with xCode, especially at the beginning of the course, since there was not much information yet for Swift4, the realization of this work and the experience I have obtained using it, have offered me great knowledge. utility, even to open some door in the world of work.



# Capítulo 1

## Introducción

La sociedad en la que convivimos hoy en día trae consigo una gran cantidad de problemas resueltos a través de las vías de los métodos informáticos y de procesos automatizados que consiguen que nuestro trabajo diario sea más eficiente, ya que es bastante más difícil que falle un proceso automático a que lo haga un humano. En otras palabras nos acercamos al mundo de Internet Of Things (IoT).

El desarrollo de este trabajo de Fin de Grado concerniente al grado de Ingeniería de Computadores, pone en marcha la idea de desarrollar una aplicación para dispositivos iOS capaz de ayudar a los usuarios que padezcan enfermedades cardiovasculares y diabetes.

A sabiendas de que el año pasado se hizo una aplicación muy parecida para dispositivos con sistema operativo Android, decidimos que sería buena idea hacerla también para los usuarios de iOS, aunque la aplicación no es la misma si que hemos partido de la idea de los compañeros que la realizaron el año pasado.

### 1.1. Motivación

Se pretende con este proyecto desarrollar una aplicación que no cumpla estrictamente con los intereses académicos, si no que sirva también de ayuda para una gran parte de la sociedad en la que vivimos. Se ha decidido, a principio de curso centrarse en la sanidad debido a varias causas:

1. Intereses personales por esta rama de la medicina en concreto, ya que algunos de mis familiares más cercanos han tenido muchos problemas de diabetes.
2. El retraso de esta materia del saber en lo que a aplicaciones móviles se refiere, si lo comparamos con otros campos profesionales.
3. La ausencia de aplicaciones que traten las necesidades de la diabetes y de los riesgos cardiovasculares en concreto. Algunos de los ejemplo quedan documentados en el tercer apartado de este capítulo: Estado del Arte.

## 1.2. Objetivos

Con el desarrollo de este trabajo de Fin de Grado se pretende poder observar cuanto antes si el usuario puede tener, o tiene, riesgos de padecer una enfermedad cardiovascular o incluso si debería tomar insulina en caso de pacientes con diabetes, así como guardar información pertinente de estos usuarios. El proyecto tiene las siguientes metas a llevar a cabo:

- Realizar un estudio acerca de los aspectos más influyentes en el sufrimiento de enfermedades relacionadas con la presión arterial o problemas del corazón, con el objetivo de poder intervenir en la evolución de este tipo de enfermedades.
- Implementar y diseñar desde cero una aplicación móvil que sea fácil de usar, atractiva a los ojos de los usuarios y que sea práctica a la hora de realizar su trabajo.
- Diseñar y confeccionar una base de datos de tipo no relacional, para la gestión de usuarios y datos ofrecidos por los mismos, así como los propios datos que genera la aplicación.
- Garantizar la autenticidad de los usuarios registrados en la aplicación, no sólo con el método estándar de autenticación por e-mail y contraseña, sino también a través de la API de la red social de Facebook.
- Poder proporcionar al usuario un cálculo de las unidades de insulina que debería tomar dependiendo de los objetivos y de las comidas que haya realizado.

## 1.3. Estado del arte

Con el objetivo de saber el impacto que pueda tener esta aplicación, se ha realizado una investigación de posibles aplicaciones que posean objetivos o rasgos similares a la de este proyecto. Esta investigación nos ha puesto en contacto con la existencia de otras aplicaciones con unos objetivos semejantes a la que se trata en este proyecto. Se nombran a continuación algunos de los competidores:

- **Appteca**

Aplicación que permite la gestión de hipertensión arterial y enfermedades cardiovasculares, este proyecto fue diseñado por la asociación española de cardiología.

<http://appteca.es>

- **PSCV**

Aplicación que permite calcular el riesgo cardiovascular dependiendo del tipo de riesgo que sufra el usuario. Esta es la más parecida a mi proyecto ya que proporciona bastante información acerca de la diabetes tipo 2. Aunque no incluye un calculador de bolos.

[https://play.google.com/store/apps/details?id=com.celmedia.minsal.pscv&hl=es\\_419](https://play.google.com/store/apps/details?id=com.celmedia.minsal.pscv&hl=es_419)

- **ASCVD Risk Estimator**

Esta aplicación tiene como objetivo el acceso a recomendaciones especiales para cada tipo de usuario, dependiendo de los cálculos que realiza con los datos que el propio usuario le proporciona. También informa a los usuarios del estilo de vida más acorde según el tipo de enfermedades que padezcan.

[https://play.google.com/store/apps/details?id=org.acc.cvrisk&hl=es\\_419](https://play.google.com/store/apps/details?id=org.acc.cvrisk&hl=es_419)

La conclusión de este pequeño estado del arte es que hay varias aplicaciones con unos objetivos parecido a la que aquí se expone, aunque la aplicación que se expone en esta memoria tiene cubiertas algunas características que anteriormente no se han mencionado. Algunas de las novedades, son por ejemplo, el cálculo del bolo prandial para los pacientes con diabetes y el cálculo del bolo corrector para después de las comidas. Así como su almacenaje para un histórico en la base de datos.





## Capítulo 2

# Especificación de la aplicación

En este capítulo se va a proceder a mostrar las principales funciones que realiza la aplicación. Las tareas más importantes que realiza son las siguientes:

- Dar de alta a los usuarios y mantener su sesión activa.
- Calcular su índice de masa corporal.
- Calcular su presión arterial.
- Calcular su colesterol.
- Calcular su adicción al tabaco.
- Calcular el bolo corrector y prandial de usuarios con diabetes.

### 2.1. Pantalla de selección del tipo de registro

Cuando ejecutamos la aplicación nos aparece una pantalla con dos botones, Figura 2.1. Uno de ellos nos permite el registro a través de la red social de Facebook, para usar este método de registro en la aplicación necesitaremos. lógicamente una cuenta de Facebook. El otro botón que se muestra en la pantalla nos llevará a un login convencional requiriéndonos un e-mail y una contraseña.

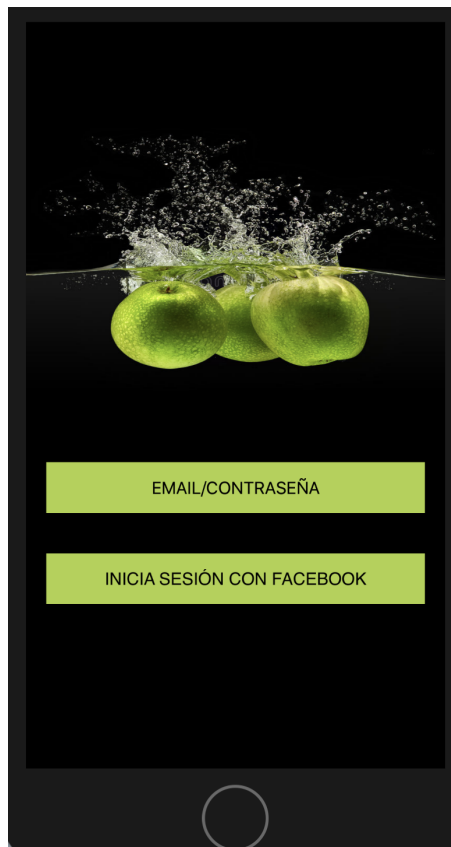


Figura 2.1: Selección de tipo de registro

## 2.2. Pantalla de login convencional

Al pulsar en el botón 'Email/Contraseña' de la vista explicada en la sección 2.1, nos llevará a este ViewController donde nos aparecerá una pantalla de login convencional pidiéndonos un usuario y una contraseña de mínimo seis caracteres, Figura 2.2. La propia API de Firebase se encarga de hacer las comprobaciones de que lo introducido en el campo e-mail, tenga este formato, y de que la contraseña tenga al menos seis caracteres. Cuando estos campos sean correctos nos llevará a la pantalla principal de la aplicación. Se incluye también en documento un diagrama de actividad de la vista, Figura 2.3

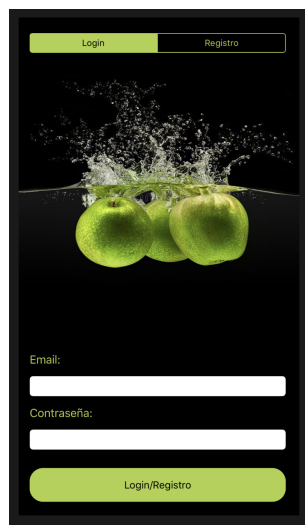


Figura 2.2: Registro Email/Contraseña

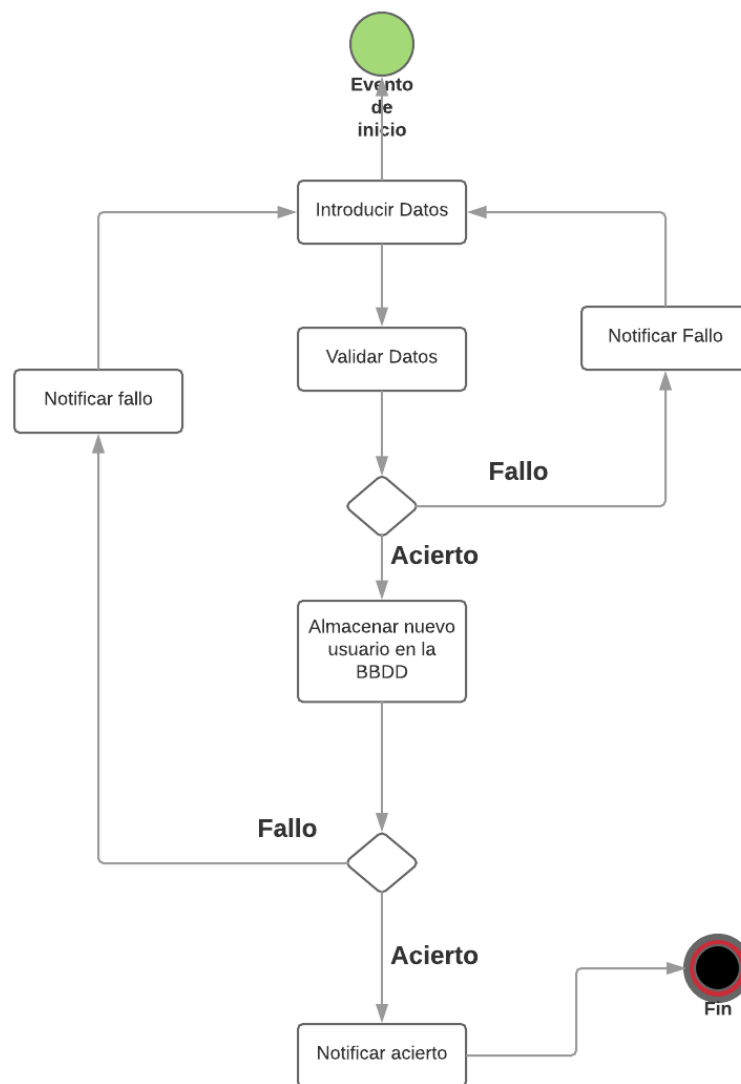


Figura 2.3: Diagrama de actividad

## 2.3. Login a través de Facebook

Al pulsar en el botón de "Login con Facebook" nos llevará a la típica pantalla de Facebook dándonos a elegir si queremos registrarnos a través de la web o a través de la aplicación de Facebook (esta opción solo aparecerá en caso de que tengas instalada su aplicación en tu dispositivo), Figura 2.4. Una vez realizado este paso nos mostrará una ventana pudiendo elegir los permisos que le ofreces a mi aplicación desde Facebook, siendo obligatorios el perfil público y el e-mail para guardarlos en la BBDD de mi aplicación. Una vez terminado el registro la aplicación te llevará a su página principal.



Figura 2.4: Pantalla de Registro a través de Facebook

## 2.4. Pantalla principal de la aplicación

Si todo el proceso de registro ha ido bien, se nos mostrará este desplegable en el que aparecen las 5 funciones principales de la aplicación: cálculo del índice de masa corporal (IMC), cálculo de la presión arterial (HTA), cálculo de tabaquismo, el calculador de bolos de insulina, y por último el calculador del colesterol. Para acceder a cualquiera de ellas solo debemos pinchar en una de las imágenes. Disponemos de un scroll para poder seleccionar cualquiera de ellas de manera intuitiva, Figura 2.5 y 2.6.

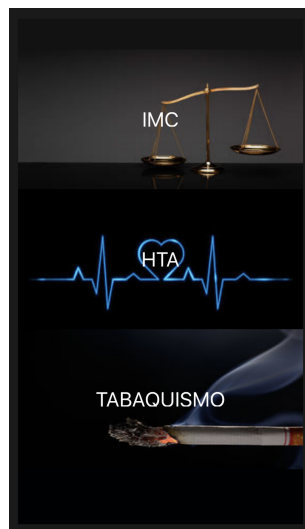


Figura 2.5: Pantalla principal



Figura 2.6: Pantalla principal

## 2.5. Calculador del índice de masa corporal

Si seleccionamos la imagen de la báscula de color dorado nos llevará a otro ViewController de la aplicación, Figura 2.7, dónde tendremos que introducir obligatoriamente los campos que se especifican, ya que todos ellos son necesarios para calcular el IMC, el cual se obtiene a través de la siguiente fórmula:

$$IMC = \frac{PESO(kg)}{ALTURA^2(cm)} \quad (2.1)$$

Pulsando en el botón de calcular nuestro IMC, nuestro peso, y nuestra altura se guardarán en la base de datos, en caso de que no hayas introducido alguno de los campos. Te saltará un mensaje de error informando que debes rellenarlos todos.

El botón de 'último IMC' sirve para que la aplicación te muestre el último IMC que calculaste, en caso de que no hayas calculado previamente un IMC, aparecerá una alerta informándote con el siguiente mensaje: "No has introducido previamente un IMC". El diagrama de actividad de esta vista se puede observar en la Figura 2.8.

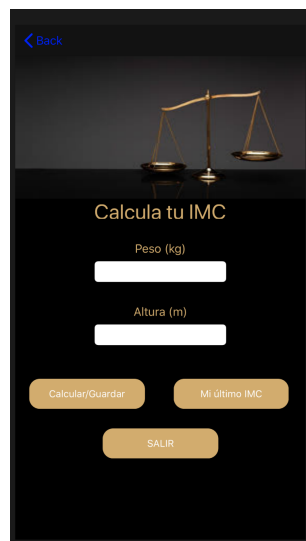


Figura 2.7: Cálculo del IMC

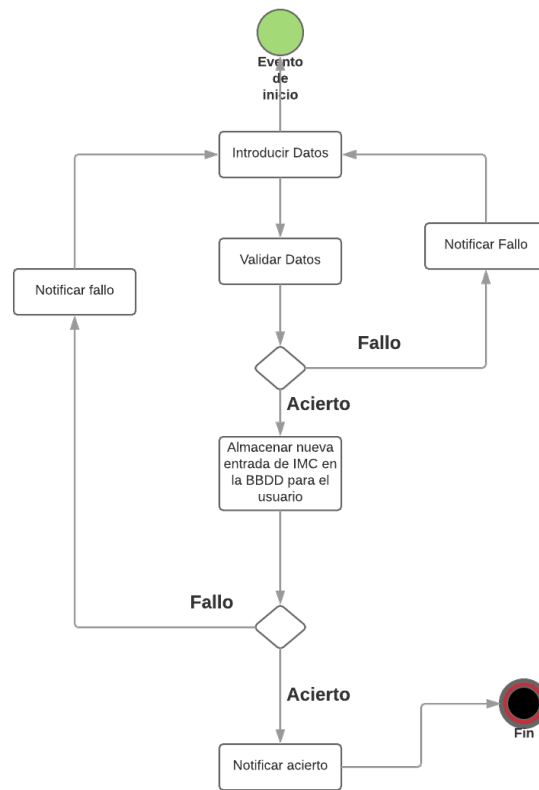


Figura 2.8: Diagrama de actividad

## 2.6. Calculador de la hipertensión arterial media

Si seleccionamos la imagen del corazón azul nos llevará a otro ViewController de la aplicación, Figura 2.9, dónde tendremos que introducir obligatoriamente los campos que se especifican, ya que todos ellos son necesarios para calcular nuestro HTA, el cual se obtiene a través de la siguiente fórmula:

$$PAM = \frac{(PAD + PAD) + PAS}{3} \quad (2.2)$$

Donde:

- PAM = Presión arterial media.
- PAS = Presión arterial sistólica.
- PAD = Presión arterial diastólica



Pulsando en el botón de calcular nuestro HTA, nuestra presión diastólica y nuestra presión sistólica, (ambas presiones han de ser introducidas en mmHG) se guardarán en la base de datos, en caso de que no hayas introducido alguno de los campos. Te saltará un mensaje de error informando que debes rellenarlos todos.

Al presionar el botón de 'Mi último HTA' la aplicación nos mostrará el último HTA que hemos calculado con nuestra sesión. En caso de que no hayamos calculado uno previamente la aplicación nos mostrará una alerta indicándonoslo. El diagrama de actividad de esta vista de la aplicación se puede ver en la Figura 2.10.



< Back



Calcula tu HTA

Presión diastólica (mmHg)

Presión sistólica (mmHg)

Calcular/Guardar

Mi último HTA

Salir

Figura 2.9: Cálculo del HTA

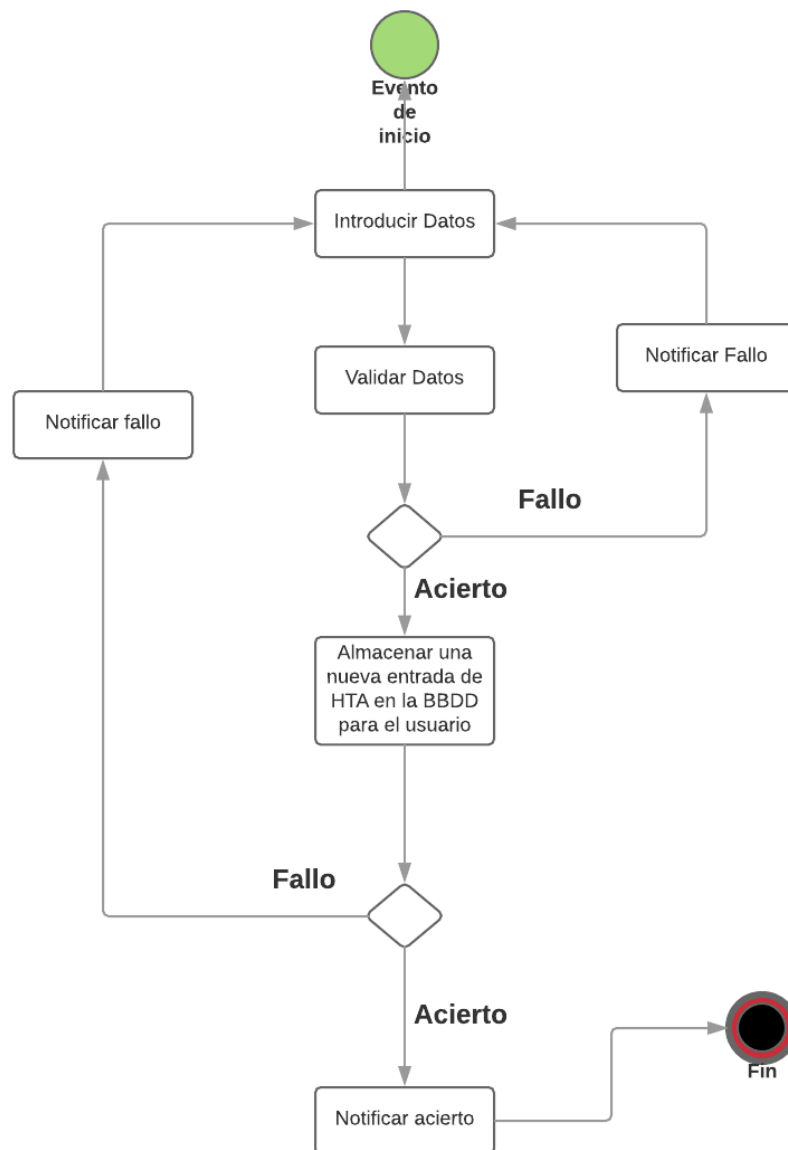


Figura 2.10: Diagrama de actividad

## 2.7. Cálculo del tabaquismo

Si seleccionamos la imagen del cigarro, la aplicación nos redirigirá a un nuevo View-Controller, Figura 2.11, dónde nos pedirá los años que llevamos fumando y los cigarrillos que fumamos cada día. Estos dos campos son obligatorios, si no los introduces no se podrá ejecutar la siguiente fórmula:

$$Tabaquismo = \frac{Años\ fumando * Cigarrillos\ diarios}{20} \quad (2.3)$$

Pulsando en el botón de calcular, se guardarán en la base de datos los datos introducidos y el tabaquismo calculado mediante la fórmula, a continuación nos saldrá una ventana emergente con la información del cálculo que se ha realizado. Si no hemos introducido todos los datos, nos saldrá otra ventana de alerta indicándonos que los datos son obligatorios.

Al presionar el botón de 'Mi último Tabaquismo' la aplicación nos mostrará el último Tabaquismo que hayamos calculado con nuestra sesión. En caso de que no hayamos calculado uno previamente la aplicación nos mostrará una alerta indicándonoslo. El diagrama de actividad de la aplicación se puede observar en la Figura 2.12.

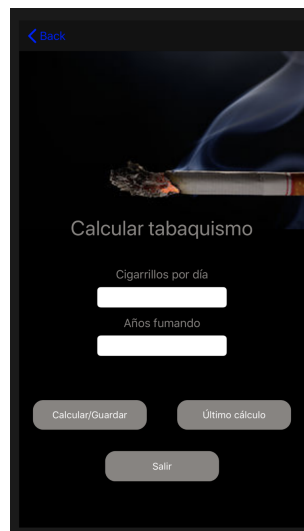


Figura 2.11: Cálculo del tabaquismo

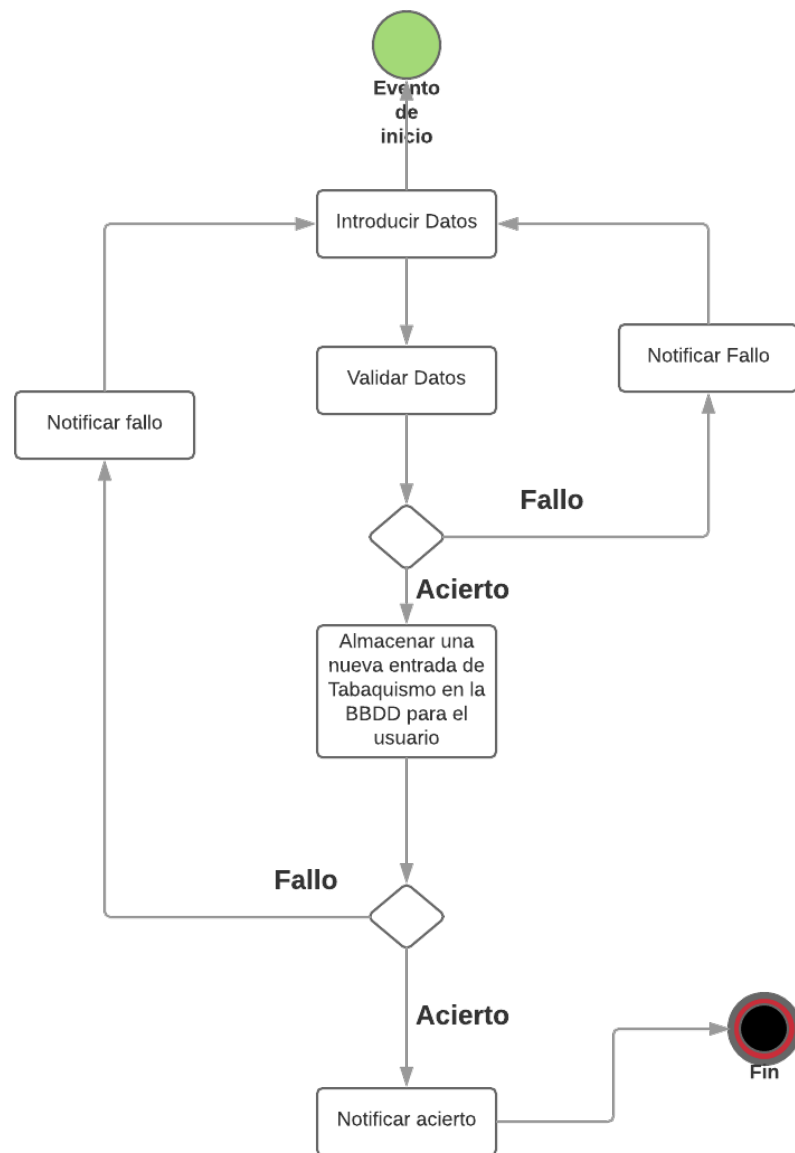


Figura 2.12: Diagrama de actividad

## 2.8. Cálculo del colesterol

Si seleccionamos la imagen del salero, la aplicación nos redirigirá a un nuevo View-Controller, Figura 2.13, dónde nos pedirá el HDL, nuestro colesterol total y la cantidad de triglicéridos. Estos tres campos son obligatorios, si no los introduces no se podrá ejecutar la siguiente fórmula:

$$LDL = ColesterolTotal - \left( HDL + \frac{Triglicéridos}{5} \right) \quad (2.4)$$

Donde:

- LDL = Lipoproteína de baja densidad.
- HDL = Lipoproteína de alta densidad.

Pulsando en el botón de calcular, se guardarán en la base de datos los datos introducidos y el LDL calculado mediante la fórmula, a continuación nos saldrá una ventana emergente con la información del cálculo que se ha realizado. Si no hemos introducido todos los datos, nos saldrá otra ventana de alerta indicándonos que los datos son obligatorios. El diagrama de actividad de esta vista está representado en la Figura 2.14.

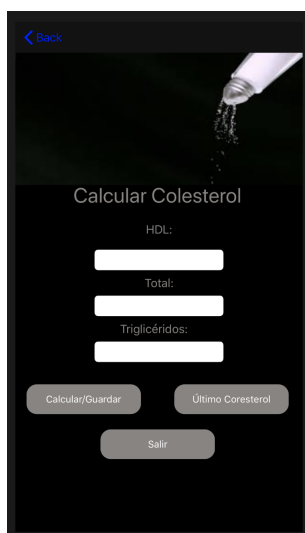


Figura 2.13: Cálculo del LDL

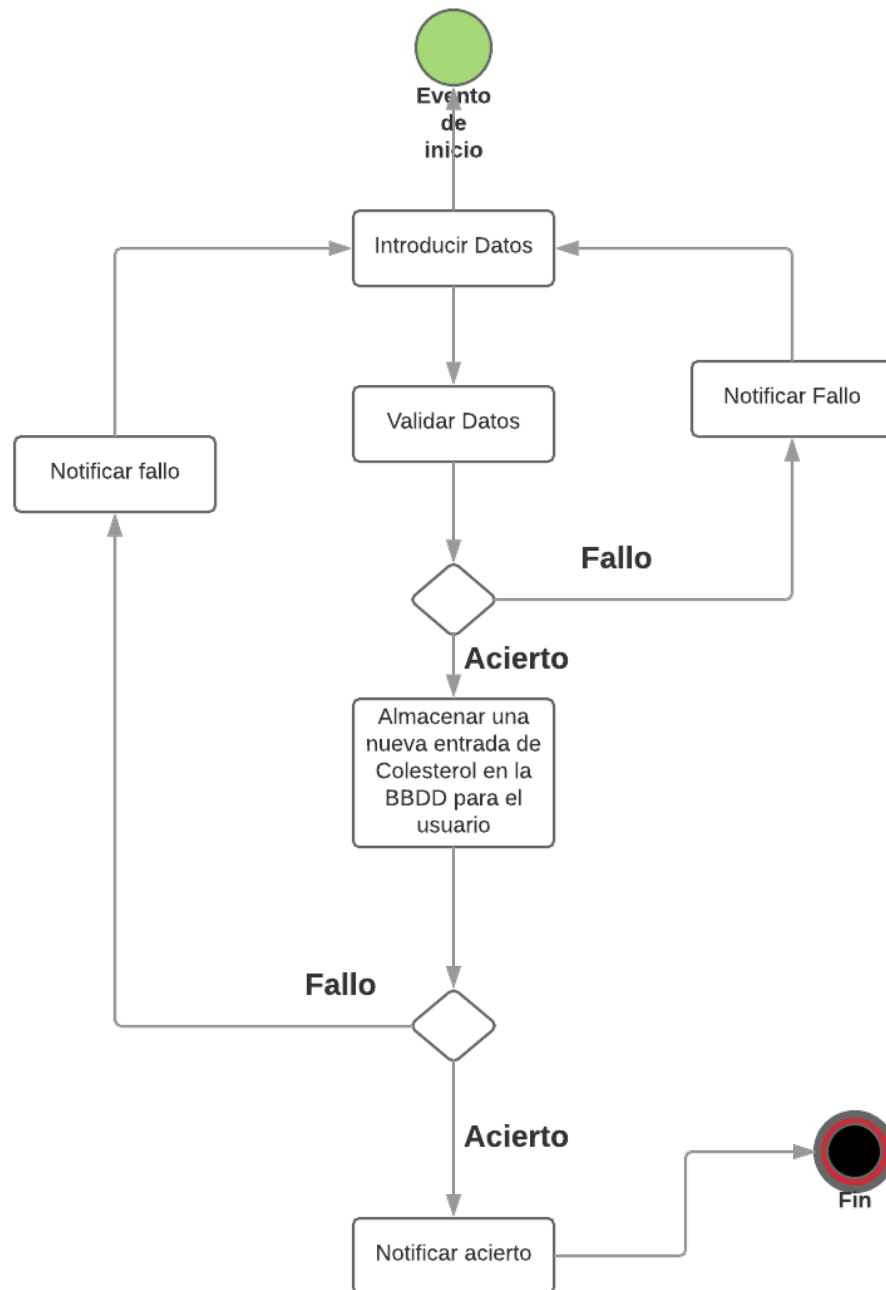


Figura 2.14: Diagrama de actividad

## 2.9. Calculador de bolos

En este apartado de la memoria vamos a proceder a detallar el funcionamiento del calculador de bolos para pacientes con diabetes de tipo I.

### 2.9.1. Introducción al calculador de bolos

En primer lugar hay que aclarar que el calculador tiene dos situaciones donde aplicarse:

- Calculador del Bolo prandial o antes de las comidas, Figura 2.16.
- Calculador del Bolo Corrector en cualquier momento del día después de una comida, Figura 2.17.

Para realizar el cálculo y por tanto la recomendación de insulina, se necesitan una gran cantidad de datos. Vamos a separarlo en dos tipos:

- Datos que deben ser proporcionados por el usuario, Figura 2.15.
- Datos que pueden ser proporcionados por el usuario, pero que idealmente se deberían calcular a partir de los datos de la base de datos o de un medidor continuo de glucosa.

### 2.9.2. Datos que deben ser proporcionados por el usuario

En la imagen de abajo, Figura 2.15, podemos ver los datos que la aplicación le va a pedir al usuario para guardarlos en la base de datos y posteriormente hacer el cálculo, tanto del bolo prandial como del bolo corrector para después de la comida.

Cuando pulsemos el botón calcular la aplicación guardará en la base de datos los datos introducidos por el usuario, esta vez no es necesario introducir los datos, ya que existen unos predefinidos, incluso podemos modificar alguno de ellos a nuestro gusto.

Si presionamos el botón de "Salir" la aplicación nos redirigirá a la pantalla de selección de login.

Figura 2.15: Pantalla para seleccionar los objetivos del usuario

### 2.9.3. Cálculo del bolo prandrial

Existen tres casos de cálculo del bolo prandrial, véase:

■ **Bolo prandrial sin necesidad de corrección:**

Supongamos que el paciente se realiza una medida de glucosa que da como resultado  $G(t) = 120mg/dl$ . Este valor se encuentra dentro del rango que se considera aceptable ya que es igual a  $G_{obj}^{pre}$ . Por lo tanto, no se debe aplicar dosis correctora. El siguiente paso es estimar el número de raciones de CH que va ingerir. Supongamos que son 6 raciones. Nos ha proporcionado el dato  $R_{\frac{I}{C}} = 1,2$ . Por lo tanto aplicando la ecuación obtenemos el número de unidades de insulina que se debe recomendar:

$$I_r(t) = 6 \cdot 1,2 = 7,2U \quad (2.5)$$

■ **Bolo prandrial con corrección positiva:**

Supongamos ahora que el paciente se realiza una medida de glucosa que da como resultado  $G(t) = 200mg/dl$ . Según los datos proporcionados por el paciente la glucosa objetivo para antes de las comidas es de 120 mg/dl. Suponiendo de nuevo una ingesta de 6 raciones de carbohidratos y aplicando la ecuación:

$$I_r(t) = C(t) \cdot R_{\frac{I}{C}} + \frac{G(t) - G_{obj}^{pre}}{S_I} \quad (2.6)$$



$$I_r(t) = 6 \cdot 1,2 + \frac{200 - 120}{40} = 9,2U \quad (2.7)$$

■ **Bolo prandial con corrección negativa:**

Supongamos ahora que el paciente se realiza una medida de glucosa que da como resultado  $G(t) = 72mg/dl$ . Según los datos proporcionados por el paciente la glucosa objetivo para antes de las comidas es de 120 mg/dl. Suponiendo de nuevo una ingesta de 6 raciones de carbohidratos y aplicando la ecuación:

$$I_r(t) = C(t) \cdot R_{\frac{I}{C}} + \frac{G_{obj}^{pre} - G(t)}{S_I} \quad (2.8)$$

$$I_r(t) = 6 \cdot 1,2 + \frac{72 - 120}{40} = 6U \quad (2.9)$$



[< Back](#)

INTRODUCIR OBJETIVOS   **BOLO PRANDIAL**   BOLO CORRECTOR

Glucosa:

Carbohidratos:

Ratio insulina-Carbohidratos:

Calcular

Figura 2.16: Bolo prandial

### 2.9.4. Bolo corrector para después de la comida

En algunas situaciones hay que volver a inyectar la insulina después de las comidas para corregir situaciones de hiperglucemia. Supongamos la situación anterior en la que habíamos echo una inyección de 6 U tras aplicar una corrección negativa. Consideremos ahora que a las 2 horas de la comida, realizamos una medida de glucosa y el resultado es  $G(t + 120) = 272mg/dl$ . Para calcular el bolo corrector usaremos:

$$I_{rc}^{post}(t + t_c) = \frac{G(t + t_c) - G_{esp}(t + t_c)}{S_I} - IOB \quad (2.10)$$



Figura 2.17: Bolo corrector

## Capítulo 3

### Arquitectura de la aplicación

El proyecto aquí expuesto es una aplicación para móviles con sistema operativo iOS que realiza las funciones de un servidor remoto en la nube, que se encarga de gestionar los accesos a la propia aplicación y a la base de datos, y un cliente. Esto se realiza a través de un Framework comprado por Google hace un par de años llamado Firebase. Todas las partes que están implicadas en la estructura de la aplicación serán explicadas en los siguientes puntos.

Como se puede observar en la Figura 3.1, se ilustra un esquema de las tecnologías que conforman la arquitectura de la aplicación.

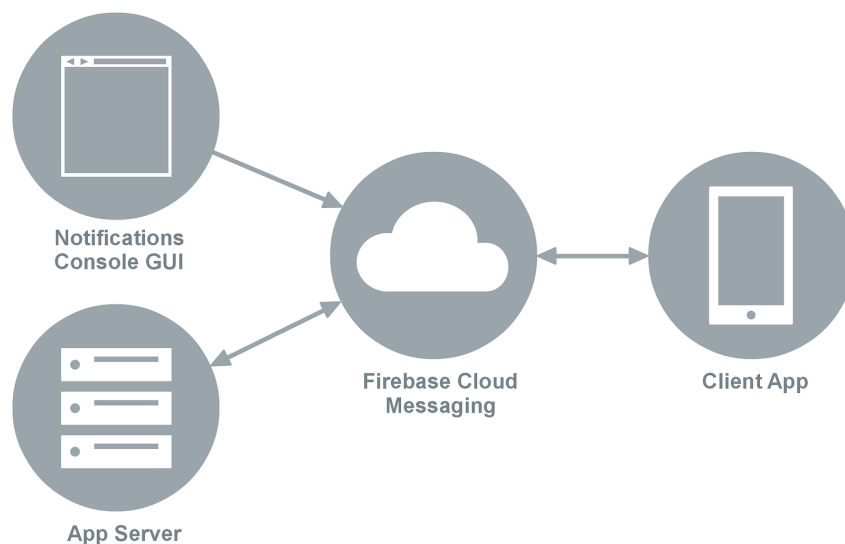


Figura 3.1: Arquitectura de la aplicación

### 3.1. Aplicación iOS

La aplicación está implementada para poder ser compatible con otras versiones del sistema operativo desde la 10.1 o superiores, pero para obtener óptimos resultados se recomienda usar la última versión de iOS (iOS 11). La aplicación está desarrollada al 90 % en Swift 4 (la versión más actual de desarrollo en los dispositivos de Apple. iOS, macOS y watchOS), sólo la parte del login con Facebook está desarrollado con Swift 3 debido a la compatibilidad en la integración con el Framework de Firebase.

### 3.2. Base de Datos

La BBDD de la aplicación se encuentra alojada en la nube de Firebase y podrá guardar en ella todos los datos fundamentales, incluidos los generados por la propia aplicación.

En la aplicación se requiere únicamente de lectura y escritura de los datos facilitados por el cliente.

En esta aplicación solamente es necesario que ejecute lecturas y escrituras en la aplicación, no necesitamos hacer modificaciones de ellos. Esta registrará información relativa a los clientes, tales como su ID de inicio de sesión, fundamental para localizar sus datos en Firebase, como los datos del imc, hta, tabaquismo, colesterol...

De esta manera al únicamente tener que escribir y leer los datos de la base de datos, en ningún caso de la modificación de los mismos, he decidido optar por una base de datos no relacional como es Firebase Realtime Database.



Figura 3.2: Logo del Framework de Firebase

### 3.3. Servidor

La función del servidor lo realiza el propio Framework de Firebase, ya que tiene disponible una nube dónde almacena todos los datos. El servidor alojado en la nube tendrá acceso a Firebase Realtime Database y a los demás servicios de Firebase como CloudStorage, hosting ...

# Capítulo 4

## Tecnologías empleadas

Presento a continuación las tecnologías empleadas para cada módulo del proyecto.

### 4.1. Aplicación iOS

Cómo en el capítulo anterior he comentado, la aplicación está realizada al 90 % con Swift 4 y el 10 % restante con Swift 3, debido a la autenticación a través de Facebook.

Cómo podemos encontrar en la página oficial de desarrolladores de Apple (<https://developer.apple.com/swift/>) esta decisión es completamente válida ya que el compilador de Swift 4 contempla los siguientes modos:

- **Modo Swift 3:** predeterminado para el código existente y construirá el código fuente que se compiló con el compilador Swift 3.
- **Modo Swift 4:** el modo Swift 4 te permite utilizar nuevas funciones y optimizaciones de rendimiento de Swift 4, algunas de las cuales pueden requerir algún tipo de migración.
- **Modo Mixto:** el utilizado en el proyecto, le permite aprovechar la interoperabilidad entre los binarios creados con el mismo compilador. Esto ayuda a los desarrolladores que tienen proyectos que mezclan paquetes escritos con Swift 3 con paquetes escritos con Swift 4, siempre y cuando todos los paquetes se hayan creado utilizando el compilador Swift

### 4.2. Base de datos

Dentro del framework de Firebase existe una base de datos en tiempo real denominada Firebase Realtime Database, la cual almacena y sincroniza datos en la base de datos

no relacional. Toda esta información se mantiene incluso cuando la aplicación no tiene conexión, además de sincronizar todos los datos a tiempo real.

Firebase Realtime Database se encuentra en un servidor en la nube. Estos datos que se recogen se sincronizan con cada cliente que se conecte a la aplicación en tiempo real. Cuando se compila la aplicación, todos los clientes que la usen van a recibir una instancia de la base de datos, en este caso sólo los usuarios de iOS, pero es perfectamente compatible con otros lenguajes y otros sistemas operativos. Voy a proceder a explicar cuales son las funciones claves de este framework:

- **Tiempo real:** cada vez que se cambian los datos en la aplicación los clientes recibirán una nueva instancia de la base de datos en milisegundos, en vez de usar las típicas consultas HTTP.
- **Sin conexión:** esta base de datos es capaz de trabajar sin conexión, ya que este framework hace que los datos de la aplicación persistan en el disco, y cuando la conexión se restablece, el cliente obtiene aquellos datos que faltaban y es capaz de sincronizarlos al estado del servidor.
- **Acceso desde cliente:** Una de las principales ventajas y uno de los motivos por lo que se ha elegido esta base de datos es por la interfaz que ofrece, es que es accesible a través de la web y no necesitas un servidor. Firebase Realtime Database se encarga de la validación y la seguridad de los datos de la aplicación, estas se basan en las expresiones al escribir o leer los datos.
- **Escalabilidad de datos:** Firebase Realtime Database, satisface la necesidad de datos de la aplicación a gran escala, dentro del mismo proyecto te da la posibilidad de separar la información en instancias dentro del mismo proyecto.

# Capítulo 5

## Modelo de datos

En este proyecto es fundamental el acceso y el tratamiento de datos, ya que sin ellos la aplicación no tendría ningún sentido. Desde el primer momento de empezar el proyecto hemos tenido claro que no íbamos a desarrollar varios roles en la aplicación, ya que iba a ser meramente informativa para el usuario de la misma.

Se utilizó el framework de Firebase para abordar el tratamiento de los datos, puesto que es perfecta para almacenar de manera horizontal y esquemática los datos de los usuarios, además se puede integrar fácilmente con otros lenguajes de programación y con otros sistemas operativos, ya que permite exportar/importar la base de datos en formato JSON.

Contamos con una lista de IDs principales que pertenecen a los usuarios registrados en la aplicación. Cuando el usuario se registra, este ID es el único dato que tenemos del mismo, y al ser una base de datos en tiempo real esta se va rellenando y haciendo más completa cuando el usuario comienza a usar la aplicación. Paso a presentar un esquema de todos los datos que puede tener un usuario:

- IDusuario
  - HTA
    - Presión diastólica: guarda la presión diastólica del usuario (mmHg).
    - Presión sistólica: guarda la presión sistólica del usuario(mmHg).
    - hta: guarda el HTA del usuario.
  - IMC
    - peso: guarda el peso en kg del usuario.
    - altura: guarda el peso en metros del usuario.
    - imc: guarda el IMC del usuario.

- Tabaquismo
  - años fumando: almacena los años que lleva fumando el usuario.
  - cigarrillos al día: guarda los cigarrillos diarios que fuma el usuario.
  - tabaquismo: guarda el Tabaquismo del usuario.
- LDL
  - Colesterol total: guarda el colesterol del usuario.
  - Hdl: almacena el HDL del usuario.
  - Triglicéridos: almacena los triglicéridos proporcionados por el usuario.
  - ldl: guarda el LDL calculado del usuario.
- Calculador de Bolos
  - Datos del calculador
    - ◇ glucosaObjetivoComidas: guarda la glucosa del usuario antes de comer.
    - ◇ glucosaObjetivoDia: guarda la glucosa objetivo del usuario durante el día.
    - ◇ glucosaObjetivoNMinutosComida: almacena la glucosa objetivo del usuario a X minutos de la comida.
    - ◇ glucosaObjetivoNoche: guarda la glucosa objetivo del usuario durante la noche.
  - Bolo prandrial
    - ◇ glucosa: guarda la glucosa introducida por el usuario
    - ◇ glucosaObjetivoPreComida: almacena la glucosa precomida guardada en el apartado "datos del calculador" del usuario
    - ◇ ratio: guarda el ratio Insulina/Carbohidratos introducido por el usuario
    - ◇ unidadesInsulina: guarda las unidades de insulina que debe tomar el usuario.
  - Bolo Corrector
    - ◇ boloCorrectorUdInsulina: unidades de insulina a corregir por el usuario
    - ◇ glucosaPostComida: almacena la glucosa de después de comer del usuario
    - ◇ tiempo: almacena el tiempo de corrección del bolo



# Bibliografía

- [1] SERGIO BECERRIL, *“Aprende a crear Apps para iPhone y iPad”*
- [2] JORGE MALDONADO B., *“Aprende Swift 4 para iOS 11”* : <https://www.udemy.com/aprende-swift-4-para-ios-11-y-lo-mejor-en-bases-de-datos/learn/v4/content>
- [3] GOOGLE INC., *“Documentación oficial de Firebase”* : <https://firebase.google.com/docs/database/ios/start>
- [4] RICHARD E. KLABUNDE, PH.D, *“Cardiovascular Physiology Concepts: Mean Arterial Pressure”*
- [5] COX NEWS SERVICE, *“Is obesity such a big, fat threat?”*, 30 de agosto de 2004
- [6] MEDLINEPLUS, *Exámenes de colesterol y sus resultados*
- [7] APPLE INC., *“Web oficial de desarrolladores”* : <https://developer.apple.com/swift>
- [8] FACEBOOK INC., *“Web oficial de desarrolladores”* : <https://developers.facebook.com/>



# Agradecimientos

A mis padres, a mis hermanos y a toda mi familia, gracias a quienes soy quien soy y hacia quienes sólo puedo expresar mi sincero agradecimiento por apoyarme durante la etapa académica que hoy culmina.

A Computadores en general, ya que sin esa complicidad y compañía diaria que nos ofrecíamos los unos a los otros esto no hubiera sido posible.

Y para finalizar, al profesor Jose Ignacio Hidalgo por haber sido mi guía durante la redacción de mi TFG.